Finally, we present a successor algorithm for the revolving door ordering, in Algorithm 2.13. In this algorithm, the successor of the last k-subset is the first one. In other words, we think of the list $A^{n,k}$ as being ordered cyclicly, and therefore we define

$$\mathsf{successor}(\{1,\ldots,k-1,n\}) = \{1,\ldots,k\}.$$

Note that this is also a minimal change.

Algorithm 2.13 begins by defining t_{k+1} to be $n+1$. This means that we do not have to handle the situation $j = k$ as a special case.

Algorithm 2.13: KSUBSETREVDOORSUCCESSOR ($\overrightarrow{T}, k, n$)

```
$t_{k+1} \leftarrow n + 1$
$j \leftarrow 1$
while ($j \le k$) and ($t_j = j$)
  do $j \leftarrow j + 1$
if $k \not\equiv j \bmod 2$
  then
    if $j = 1$
      then $t_1 \leftarrow t_1 - 1$
      else
        $t_{j-1} \leftarrow j$
        $t_{j-2} \leftarrow j - 1$
  else
    if $t_{j+1} \ne t_j + 1$
      then
        $t_{j-1} \leftarrow t_j$
        $t_j \leftarrow t_j + 1$
      else
        $t_{j+1} \leftarrow t_j$
        $t_j \leftarrow j$
return ($\overrightarrow{T}$)
```

2.4 Permutations

2.4.1 Lexicographic ordering

We now look at the generation of all $n!$ permutations of the set $\{1,\ldots,n\}$. A *permutation* is a bijection from a set to itself. One way to represent a permutation $\pi : \{1,\ldots,n\} \to \{1,\ldots,n\}$ is by listing its values, as follows:

$$[\pi[1],\ldots,\pi[n]].$$

We call this the *list representation* of the permutation π. Saying that π is a permutation is equivalent to saying that each element in $\{1,\ldots,n\}$ occurs exactly once in this list.

First, we will look at the lexicographic ordering of permutations. The lexicographic ordering is defined in terms of the list representation. As an example, when $n = 3$, the lexicographic ordering of the six permutations of $\{1,2,3\}$ is as follows:

$$[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1].$$

We begin by describing an algorithm for generating permutations in lexicographic order. This generation algorithm depends on a successor algorithm that finds the permutation that immediately follows a given permutation (in lexicographic order). In Algorithm 2.14, π is a permutation of $\{1,\ldots,n\}$ given in list representation.

Algorithm 2.14 has four steps. In the first **while** loop, we find i such that

$$\pi[i] < \pi[i+1] > \pi[i+2] > \cdots > \pi[n].$$

Note that by setting $\pi[0]$ to 0, we ensure that the **while** loop terminates with $0 \le i \le n-1$. If $i = 0$, then

$$\pi = [n, n-1, \ldots, 1]$$

is the last permutation lexicographically and has no successor. Otherwise, we proceed to the second **while** loop, where we find the integer j such that $\pi[j] > \pi[i]$ and $\pi[k] < \pi[i]$ for $j < k \le n$ (i.e., j is the position of the last element among $\pi[i+1],\ldots,\pi[n]$ that is greater than $\pi[i]$). The third step is to interchange $\pi[i]$ and $\pi[j]$, and the fourth step is to reverse the sublist

$$[\pi[i+1],\ldots,\pi[n]].$$

Algorithm 2.14: PERMLEXSUCCESSOR (n, π)

```
$\pi[0] \leftarrow 0$
$i \leftarrow n - 1$
while $\pi[i+1] < \pi[i]$
  do $i \leftarrow i - 1$
if $i = 0$
  then return ("undefined")
$j \leftarrow n$
while $\pi[j] < \pi[i]$
  do $j \leftarrow j - 1$
$t \leftarrow \pi[j]$
$\pi[j] \leftarrow \pi[i]$
$\pi[i] \leftarrow t$
for $h \leftarrow i + 1$ to $n$
  do $\rho[h] \leftarrow \pi[h]$
for $h \leftarrow i + 1$ to $n$
  do $\pi[h] \leftarrow \rho[n + i + 1 - h]$
return ($\pi$)
```